|  |
| --- |
| Module - React - Components, State, Props |

|  |
| --- |
|  |

**(Q-01) What is ReactJS?**

- React.js is a front-end JavaScript framework developed by Facebook.

- To build composable user interfaces predictably and efficiently using declarative code, we use React.

- It’s an open-source and component-based framework responsible for creating the application’s view layer.

- ReactJs follows the Model View Controller (MVC) architecture, and the view layer is accountable for handling mobile and web apps.

- React is famous for building single-page applications and mobile apps.

**(Q-02) What is NPM in React Js ?**

- npm is an abbreviation used for the node package manager.

- It is a package manager for JavaScript.

- It is the default package manager that comes with NodeJS when you install it.

- It consists of a command-line interface and an online database of public packages and private packages that are paid which iscalled the npm Registry.

**(Q-03) What is Role of Node Js in react Js?**

1. Scalability

The combination of Node and React allows developers to build dynamic, large, data-driven web apps that are responsive across multiple devices. It would help if you had scalability while working on large projects and maintaining your website’s performance.

2. MERN Stack

MERN Stack means MongoDB, Express, React, and Node.js. And no better combo than these frameworks can provide a unique dimension to a website. You can use Node.js with MERN Stack alongside React.js. Hence, you can use the Node and React combination for web app development.

3. JSON APIs

Creating JSON (JavaScript Object Notation) APIs for web development is competent due to the high code reusability and access to immediate code sharing in React.js. And Node.js can effectively allow this.

4. Real-Time Data

If your business app handles real-time data management or aims to build a data streaming app, you should use Node.js as your app requires an ongoing server connection.

5. Fast Development When using the combination of React and Node for web app development, you can receive a high ROI and save your money and time. After all, these technologies work excellently to offer an effective platform for building rapid functioning and easy-to-maintain websites.

6. SPAs (Single Page Applications)

Does your business require a single-page app with asynchronous data loading? Then, you must select React with Node back-end as it allows building a lightweight back-end model through callback functions.

7. Single Language for Front-end and Back-end

With the combination of Node and React, developers don’t require learning complex back-end languages like Python or Ruby. They can use Node for server-side development and React for front-end code building without switching between frameworks and programming languages. And it saves resources, money, and time.

8. High Server Load

The combination of Node.js and React can balance the high server requests and load when developers work on web app development.

9. Organized Process

The deadly combination of React and Node create an organized web development process. After all, these technologies are scalable, effective, and fast. When working together, they can help you build high-functioning websites.

10. Increased Use of JavaScript

**(Q-04) What is CLI command In React Js?**

- React have its own CLI but currently they are only supporting creating an app (create-react-app).

- create-react-app used to generate the boilerplate version of a React application thru command line.

- npx create-react-app my-app or npm create-react-app my-app

- create-react-app has taken care of setting up the main structure of the application as well as a couple of developer settings.

- Most of what you see will not be visible to the visitor of your web app.

- React uses a tool called webpack which transforms the directories and files here into static assets.

- Visitors to your site are served those static assets.

**(Q-05) What is Components in React Js?**

- Components are independent and reusable bits of code.

- They serve the same purpose as JavaScript functions, but work in isolation and return HTML.

- Components come in two types, Class components and Function components, in this tutorial we will concentrate on Function components.

Function Component

- Here is the same example as above, but created using a Function component instead.

- A Function component also returns HTML, and behaves much the same way as a Class component, but Function components can be written using much less code, are easier to understand, and will be preferred in this tutorial.

Example

- Create a Function component called Car

- function Car() {

return <h2>Hi, I am a Car!</h2>;

}

Class Component

- A class component must include the extends React.Component statement. This statement creates an inheritance to React.Component, and gives your component access to React.Component's functions.

- The component also requires a render() method, this method returns HTML.

Example

- Create a Class component called Car

- class Car extends React.Component {

render() {

return <h2>Hi, I am a Car!</h2>;

}

}

**(Q-06) What is Header and Content Components in React Js?**

Header Component:

- The Header component typically represents the top section of a web page or an application.

- It often contains elements like the site's logo, navigation menus, user profile information, and any other content that you want to display consistently at the top of the page.

- The Header component is reusable across different pages or sections of your application, allowing you to maintain a consistent look and feel throughout.

Example

import React from 'react';

const Header = () => {

return (

<header>

<nav>

<ul>

<li><a href="/">Home</a></li>

<li><a href="/about">About</a></li>

<li><a href="/contact">Contact</a></li>

</ul>

</nav>

</header>

);

};

export default Header;

Content Component:

- The Content component is a more generic term used to describe the main section of a web page or an application.

- It's where the primary content or functionality of the page is displayed.

- This could include text, images, forms, lists, or any other type of content that makes up the core of the page's purpose.

- Content components are often designed to be flexible and reusable, allowing you to swap out different content while keeping the same layout and structure.

import React from 'react';

const Content = () => {

return (

<div className="content">

<h1>Welcome to Our Website</h1>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>

{/\* Additional content elements \*/}

</div>

);

};

export default Content;

**(Q-07) How to install React Js on Windows, linux Operating System? How to install NPM and How to check version of NPM?**

Installing React.js and npm on Windows:

- Install Node.js: React.js requires Node.js, which includes npm. You can download the Windows installer from the official Node.js website: https://nodejs.org/

- Visit the website and download the LTS (Long Term Support) version for Windows.

- Run the installer and follow the installation instructions.

Verify Installation:

- After the installation is complete, open a Command Prompt or PowerShell window and enter the following commands to verify that Node.js and npm have been installed successfully:

- node -v

- npm -v

Create a React App:

- To create a new React application, you can use the following command in the Command Prompt or PowerShell:

- npx create-react-app my-react-app

-

- This will create a new directory named my-react-app containing a basic React project.

- Navigate to the App Directory: Use the following command to navigate to the app directory:

- cd my-react-app

Start the Development Server: Start the development server by running:

- npm start

- This will start the React development server, and you can access your app in a web browser at http://localhost:3000.

Installing React.js and npm on Linux:

Install Node.js:

- Open a terminal and use the following commands to install Node.js and npm using a package manager appropriate for your Linux distribution. Here, we'll use apt for Ubuntu/Debian-based systems and dnf for Fedora:

For Ubuntu/Debian:

- sudo apt update

- sudo apt install nodejs npm

- For Fedora:

- sudo dnf install nodejs npm

- Verify Installation: After installation, verify that Node.js and npm are installed by running:

- node -v

- npm -v

Create a React App:

- Follow the same steps as mentioned for Windows to create and navigate to a new React app.

- Start the Development Server: Similarly, start the development server using:

- npm start

- Remember that package manager commands (like apt or dnf) might require administrative privileges, so use sudo as needed.

- Checking the Version of npm:

- To check the version of npm, simply open a terminal and run:

- npm –v

**(Q-08) How to check version of React Js?**

- three ways to find out the React version.

- Using package.json file

- Using command line

- Using the version property of default import from React

Using package.json file:

- The package.json contains metadata about our project.

- It is created by default when we create our React project.

- We can create a react app using the command mentioned below.

- npx create-react-app name\_of\_the\_app

Using the command line:

- We can easily check the React version by using the command mentioned below on our command line.

- npm view react version

Using the version property of default import from React:

- The default import from React library is an object that has a version property on it.

- We can use this property inside our JSX elements in our desired manner.

- import React from 'react';

- let a = React.version

**(Q-09) How to change in components of React Js?**

1. Update in state:

- The state change can be from a prop or setState change to update a variable(say).

- The component gets the updated state and React re-renders the component to reflect the change on the app.

2. Update in prop:

- Likewise the change in prop leads to state change and state change leads to re-rendering of the component by React.

3. Re-rendering of parent component:

- Whenever the components render function is called, all its subsequent child components will re-render, regardless of whether their props have changed or not.

**(Q-10) How to Create a List View in React Js?**

Step 1:

- Create a list of elements in React in the form of an array and store it in a variable.

- We will render this list as an unordered list element in the browser.

## Step 2:

- We will then traverse the list using the JavaScript map() function and updates elements to be enclosed between <li> </li> elements.

## Step 3:

- Finally we will wrap this new list within <ul> </ul> elements and render it to the DOM.

## Example..

import React from 'react';

import ReactDOM from 'react-dom';

const numbers = [1,2,3,4,5];s

const updatedNums = numbers.map((number)=>{

return <li>{number}</li>;

});

ReactDOM.render(

<ul>

{updatedNums}

</ul>,

document.getElementById('root')

);